

REMARKS

This Amendment is filed in response to the Office Action mailed February 23, 2009. All objections and rejections are respectfully traversed.

Claims 1-5 and 30-67 are in the case.

New claims 66-67 have been added.

Claims 1, 31-32, 38-39, 45, 51-52, 58-61, and 64-65 have been amended.

Interview Summary

Applicant would like to thank Examiner Colan for conducting the Applicant Initiated Interview on April 6, 2009 and for helping to advance this Application closer to allowance. Generally, the issue discussed involved the definition of a file handle and whether or not an HTML text page was similar to a file handle. With this discussion in mind, the limitations of the claims were compared to the cited prior art. As a result of the interview, Examiner stated that she would take a closer look at Applicant's argument. As requested by Examiner, Applicant has expanded the Amendment to include greater detail of the differences between an HTML text page and Applicant's file handle in the interest of moving prosecution forward.

Rejections Under 35 U.S.C. § 103

At Paragraph 7 of the Office Action, claims 1-2, 4-5, 30-34, 36-40, 42-47, 49-53, 56-60, 62-64, and 65 were rejected under 35 U.S.C. 103(a) as being anticipated by Chandrashekhar et al., U. S. Patent Publication 2005/0033988 published on February 10, 2005 (hereinafter "Chandrashekhar"), and in view of Gvily, U.S. Patent Application Publication No. 2002/0078201 published on June 20, 2002 (hereinafter "Gvily").

Applicant's claimed novel invention, as set out in representative claim 1, comprises in part:

1. A method for establishing identity in a file system, comprising:

receiving a file request concerning an indicated file from a client,
the request received by a proxy;
forwarding the request from the proxy to a file server;
returning a reply associated with the file request from the file
server to the proxy;
inserting, by the proxy, metadata into a file handle; and
sending, by the proxy, the file handle with the metadata inserted
in the file handle to the client, the metadata to be used in further
requests to identify the client and the indicated file.

Chandrashekhar discusses processing file requests sent by a client and received by a proxy using security applications to encrypt, decompress, verify, and decrypt network data by a server receiving the files from the proxy [0058; 0071]. Header policy information is determined, generated, and then stored on the filer server [0055; Fig. 4-5]. Metadata of the header policy information then being sent by the server back to the proxy is *stripped from the file* (by the proxy) *before the file is returned to the client* [0038; Fig. 8; 0055; 0069-0070].

Gvily discusses embedding data in a *text page*, but more particularly, embedding either metadata or scripts into Hypertext Markup Language (*HTML*) (*text*) *pages*. HTML is the source code language used to create web pages. In operation, an intermediary proxy analyzes the unstructured *source code* data of the *HTML* (*text*) *pages*, then attempts to understand the meaning behind the unstructured *source code* data, then associates metadata with some of the unstructured *source code* data, and then stores this metadata back into the original *source code* of the *HTML* (*text*) *page* [0018; 0005]. The metadata of understood data will be embedded into the *source code* of the *HTML* (*text*) *page*, effectively *altering the source code* [0035-0036; see also Figs. 4-5].

Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Gvily, does not disclose Applicant's claimed novel use of **inserting, by the proxy, metadata into a file handle** and then **sending the file handle with the**

metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client and the indicated file.

Applicant claims ***inserting, by the proxy, metadata into a file handle***. Broadly stated, Network File System (NFS) supports a lookup procedure, which converts a filename into a *file handle*. For example, NFS uses the *file handle* (e.g., a unique, immutable identifier, usually an inode number, or disk block address) to uniquely identify exported files. In the example, when a client requests to access a file, the server constructs a file handle that identifies the requested file, and the identifier (file handle) is used in communications between the server and the client to access that requested file. With that being said, after ***inserting metadata into the file handle***, Applicant further claims ***sending the file handle with the metadata inserted in the file handle to the client***. As such, ***the metadata*** may subsequently be used in further requests to ***identify the client and the indicated file***.

As stated above and stated by Examiner in the Office Action, Chandrashekhar *removes the metadata* from the file data/file attributes (e.g., file handle) *before* returning the file to the client, and thus does not show Applicant's claimed ***sending, by the proxy, the file handle with the metadata inserted in the file handle to the client***. More importantly, Chandrashekhar explicitly *teaches away* from Applicant's claimed ***sending the file handle with the metadata inserted in the file handle to the client*** since Chandrashekhar *removes the metadata* from the file data/file attributes *before* returning the file to the client as a safety measure. As such, because Chandrashekhar removes the metadata before returning the file to the client, there would be no motivation to combine Chandrashekhar with any potential reference. As a result, in addition to Chandrashekhar being totally silent to Applicant's claimed ***sending, by the proxy, the file handle with the metadata inserted in the file handle to the client***, Chandrashekhar is also legally precluded from being used as a reference under 35 U.S.C. §103 as Chandrashekhar explicitly *teaches away* from Applicant's claimed invention.

As noted above, Gvily shows embedding metadata into the *source code* of a Hypertext Markup Language (*HTML*) (*text*) *page*. Applicant respectfully argues that embedding metadata into the *source code* of an *HTML* (*text*) *page* is not the same as Applicant's claimed **inserting metadata into a file handle**. Specifically, in one example, embedding metadata into the *source code* of an *HTML* (*text*) *page* would require altering the *HTML* (*text*) *page's source code*, as is stated by Gvily at paragraph 36. This is because *HTML text* is the *source code* language used to enable the web pages themselves to be more animated and interactive. In contrast, Applicant's **inserting metadata into a file handle** does not require altering the requested file. Put another way, the *source code* of an *HTML* (*text*) *page* is not the same as a *file handle*.

To further clarify the distinction between the *source code* of an *HTML* (*text*) *page* and Applicant's claimed *file handle*, Applicant respectfully directs Examiner to the following reference (<http://www.yourhtmlsource.com/starthere/whatishtml.html>) which is cited below, in relevant part:

What is HTML?

HTML is a computer *language* devised to allow website **creation**. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn...and quite powerful in what it allows you to **create**...

The definition of HTML is **HyperText Markup Language**.

- *HyperText* is the method by which you move around on the web — by clicking on special text called **hyperlinks** which bring you to the next page...
- *Markup* is what **HTML tags** do to the text inside them. They mark it as a certain type of text (*italicized* text, for example).
- **HTML** is a *Language*, as it has code-words and syntax like any other language.

How does it work?

HTML consists of a series of short **codes** typed into a text-file by the site author — these are the tags. The text is then **saved as a html file**, and **viewed through a browser**, like *Internet Explorer* or *Netscape Navigator*. This browser reads the file and translates the text into a visible form... You can use anything from a rudimentary text-editor to a powerful graphical editor to **create HTML pages**. (emphasis added)

To reiterate, HTML is a *source code* language used to *create web pages*. In contrast, Applicant's claimed *file handle* is, for example, a structure (e.g., a unique, immutable identifier, usually an inode number, or disk block address) used to uniquely identify exported files. Therefore, Gvily does not show Applicant's claimed **inserting metadata into a file handle** because an *HTML (text) page* is not a *file handle*, and as such, inserting metadata into the *source code* of an *HTML (text) page* is not the same as inserting metadata into a *file handle*.

However, even if inserting metadata into the *source code* of an *HTML (text) page* was the same as inserting metadata into a *file handle*, Gvily still does not show Applicant's claimed **metadata used in further requests to identify the client and the indicated file**. Specifically, as noted above, the newly embedded metadata in Gvily's HTML *source code* is used to alter the HTML *source code* and change how a website is viewed through a web browser. Applicant respectfully argues that altering *source code* and changing how a website is viewed does not allow the web page or altered *source code* to **identify** a requesting *client* and/or **the indicated file**. Particularly, if one assumes Gvily's website is the requested "file", then it is the web address that identifies the website, and not the HTML *source code*. Further, neither the website nor the altered HTML *source code* **identify** a requesting *client*. Thus, even if inserting metadata into the *source code* of an *HTML (text) page* was the same as inserting metadata into a *file handle*, Gvily is still silent to Applicant's claimed **metadata used in further requests to identify the client and the indicated file**.

Accordingly, Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Gvily, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. § 103. Chandrashekhar and Gvily, taken singly or in any combination, does not disclose Applicant's claimed novel **inserting, by the proxy, metadata into a file handle and then sending the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client and the indicated file.**

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art references was derived, in part, from the following excerpts:

Chandrashekhar

[0038]...The meta-data relates to key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, integrity checks for file data. The meta-data is stripped off before the file data/file attributes are returned to the client... (emphasis added)

Gvily

[0018] The invention described hereinafter discloses a computer implemented method for embedding data in a text page at a proxy, where the proxy is generally an intermediary between a resource and a request for a resource. More particularly, the invention provides exemplary systems and methods for embedding either meta-data or scripts into HTML pages by means of a proxy. The intermediary proxy analyzes the unstructured data of a hypertext page, understands the meaning behind the data, associates meta-data with some of the unstructured data and stores this meta-data back into the original hypertext page. The invention potentially stores meta-data in a location that is hidden from the user's view so that it is unobtrusive but easily retrievable. (emphasis added)

[0034] Embedding Data Examples

[0035] FIG. 4 shows an example of a prior art hypertext page 401 consisting of some text and a hyperlink 402. The following illustrates source code that may be used to render that page:

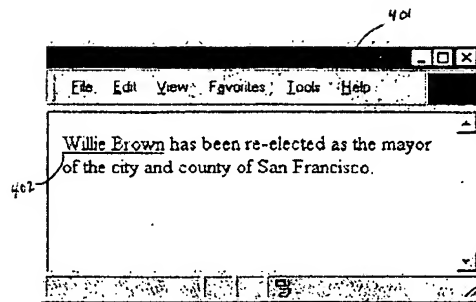


Figure 4 - Prior Art

```
1 <HTML> <HEAD> </HEAD> <BODY> <A
  HREF="http://xyz.somewhe- re.com">Willie Brown</A> has been re-
  elected as the mayor of the city and county of San Francisco. </BODY>
</HTML>
```

[0036] For purposes of the example, assume that the analysis process for the HTML page has recognized two objects on the page: Willie Brown and San Francisco. Willie Brown is recognized as a name, Willie as a first name and Brown as a last name. Continuing, San Francisco is recognized as a location. The meta-data of these objects will be embedded into the web page, effectively altering the source code to something like:

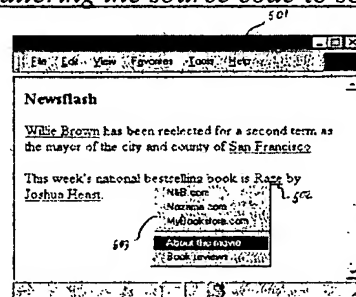


Figure 5

```
2 <HTML> <HEAD> </HEAD> <BODY> <A
  HREF="http://xyz.somewhere.com"
  META="<PERSON><FIRST>Willie</FIRST>
  <LAST>Brown</LAST></PERSON>">Willie Brown</A> has been re-
  elected as the mayor of the city and county of <SPAN
  META="<LOCATION><CITY>San
  Francisco</CITY><STATE>CA</STATE></LOCATION>">- San
  Francisco</SPAN>. </BODY> </HTML>
```

(emphasis added to [0035 and 0036])

Rejections Under 35 U.S.C. § 103

At Paragraph 8 of the Office Action, claims 3, 35, 41, 48, 54-55, and 61 were rejected under 35 U.S.C. 103(a) as being unpatentable over Chandrasekhar in view of Gvily, and in further view of Ohazama et al., U. S. Patent No. 7,225,207 issued on May 29, 2007 (hereinafter "Ohazama").

Applicant respectfully notes that claims 3, 35, 41, 48, 54-55, and 61 are all dependent claims, and these dependent claims are dependent from independent claims which are believed to be in condition for allowance. Accordingly, claims 3, 35, 41, 48, 54-55, and 61 are believed to be in condition for allowance.

New Claims

New claims 66-67 were added and are believed to be in condition for allowance.

Applicant's claimed novel invention, as set forth in representative new claim 66, comprises in part:

66. (New) A method, comprising:
receiving, by a proxy, a file request for a file sent from a client;
forwarding the file request from the proxy to a file server;
returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle;
inserting, by the proxy, metadata into the file handle;
sending, by the proxy, the file handle with the metadata inserted in the file handle to the client; and
using, by the client, the metadata inserted into the file handle in a subsequent file request to identify the client and the file.

As discussed above, Chandrasekhar, taken singly or in any combination with Gvily, does not disclose Applicant's claimed novel use of **inserting, by the proxy, metadata into a file handle** and then **sending the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client and the indicated file**. As such, Chandrasekhar, taken singly or in any combination with Gvily, is legally insufficient to render new claims 66-67 obvious under 35 U.S.C. § 103.

Conclusion

All new claims and claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/
Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927